

An Adaptive Method for Evaluating Multidimensional Contingent Claims. Part I

Lars O. Dahl

15th December 2001

Department of Mathematics, Norwegian University of Science and Technology, N-7491 Trondheim,
Norway and Storebrand Investments, PO Box 1380, N-0114 Oslo, Norway (correspondence address)
Email: lars.oswald.dahl@storebrand.com
Url: <http://www.math.ntnu.no/~larsosw>

Abstract The paper presents an adaptive method for the evaluation of multidimensional integrals over the unit cube. The measure used to partition the domain is suited for integrands which are monotonic in each dimension individually, and is therefore suitable for problems stemming from finance where this is often the case. We use a QMC method for each sub-problem resulting from the partitioning of the domain. The article is part one of a work on this topic, and presents the method together with various local variance reduction techniques. The material is presented with an alignment to option pricing problems. In the companion paper we present an option pricing problem and simulation results on different setups of this. We compare the convergence properties of the adaptive method with the convergence properties of the QMC method used directly on the problem. We find that the adaptive method in many configurations outperform the conventional QMC method, and we develop criteria on the problem for when the adaptive method can be expected to outperform the conventional.

1 Introduction

The fair value of contingent claims can be expressed as an expected value, which in turn can be written as an integral. In many situations these integrals are multidimensional. The value of the integrand can vary significantly in the domain of integration, and in some cases there can be only small parts of the domain in which the integrand is non-zero. Since Monte Carlo methods and QMC methods distribute the evaluation points as evenly as possible, they will waste calculations on regions which are not important. An adaptive method tries to allocate the resources to the important parts of the domain.

In option pricing problems where the option is written on several underlying assets and/or the option price is path dependent in time, the price is found by calculating a multidimensional integral. QMC methods are often deployed for these problems, and additional methods for reducing the variance of the estimator for the price of the option can be found. The adaptive technique incorporates some common variance reduction principles implicitly, but the modifications of the integrand are not exactly the same. In particular the adaptive method is closely related to importance sampling and stratified sampling. References on these methods are [15], [14], [16], [17], [22], [25], [26]. The purpose of these methods, as the adaptive, is to use the samples from the QMC generator in a more effective way than distributing them evenly in the domain. Some knowledge of

the integrand must be present, or collected, for the success of these methods. Importance sampling can be dangerous in that one is not guaranteed lower variance. The variance can actually blow up for special cases. The advantage of the adaptive method is that information is collected as the calculation is done, and function evaluations of the integrand is used to guide the distribution of the points in the domain.

The challenge in getting the adaptive method to perform well is to find good measures of the behavior of the integrand in local domains, and to find a data structure that minimizes overhead from the process of analyzing the domain. The basic algorithmic approach in the article is partly based on the work presented in [1], but the approach is modified in some essential aspects concerning the use of function evaluations in local domains. In addition to presenting the principles of the adaptive method, we develop and formulate variance reduction techniques applied to the local domains produced by the sub-division process of the adaptive method. This, together with the use of MC and QMC for the evaluation of the sub-problems resulting from the adaptive process is to our knowledge not investigated before.

In the companion paper [6] we perform numerical tests and find criteria for deciding when the adaptive method can be expected to perform well.

2 QMC integration

The goal is to evaluate multidimensional integrals over the region $\Omega = [0, 1]^D$ accurately, effectively and robust. When $D > 4$ the MC and QMC methods are competitive with any advanced numerical integration method for general integrands. For large D they are the only realistic alternatives. The estimator most commonly used for the integral is given by

$$F = \int_{\Omega} f(x) dx \approx \frac{1}{L} \sum_{l=1}^L f(y_l), \quad (1)$$

where $\{y_l\}$ is a sequence of vectors from a pseudo random number generator or a low discrepancy sequence. The MC method is based on drawing random or pseudo random numbers as arguments to f , while the QMC methods use sequences of arguments to f designed to be as evenly distributed as possible in Ω . In this article we use the Mersenne Twister as pseudo random number generator and an extension of the Halton¹ sequence as basis for the QMC method. References on low discrepancy sequences are e.g. [20], [24], [23].

¹ We have in previous work ([8] and [9]) found that the Halton leaped sequence with leap number 31, presented in [19], performs well compared to other easily implemented sequences

3 The adaptive method

It is easy to construct a QMC estimator for the integral over a part of Ω , and we therefore can construct a method to evaluate the integral over all of Ω as a sum of such estimated values. Let $\Omega = \cup_i \omega_i$, $\cap_i \omega_i = \emptyset$, $i = 1, \dots, P$. Then

$$F = \sum_{i=1}^P \int_{\omega_i} f(x) dx \quad (2)$$

$$\approx \sum_{i=1}^P \frac{|\omega_i|}{\Delta L_i} \sum_{l=L_{i-1}}^{L_i-1} f(y_l), \quad (3)$$

where $\Delta L_i = L_i - L_{i-1}$ and y_l is scaled such that $y_l \in \omega_i$ when $l \in [L_{i-1}, L_i]$. $|\omega_i|$ is to be understood as the volume of ω_i . The adaptive algorithm should decide on the number of sub-domains and their sizes. That is P and $|\omega_i|$, $\forall i$. Furthermore, the algorithm has to pick the best set of sub-domains, and how many simulation points ΔL_i to use in each of them. Alternative approaches use information from the integrand to develop approximations of the integral in sub-domains with a deterministic approach rather than with simulation. This is done in e.g. [1], [13], [3], [25].

We have chosen to use a binary tree to represent the domain decomposition. Each node in the tree corresponds to a distinct part of the domain, and when we expand the tree we divide the domain represented by a node in two parts (not necessarily of equal size). The criteria we use to decide on division are twofold: First we find an estimate of how much the domain contributes to the overall variance of the integral, and if this is more than a preset amount we split the domain. In order to avoid an explosive increase in partitions, we only create two new subdomains from each domain that meets the variance criterion. They are produced by dividing the domain across the axis corresponding to the dimension along which the integrand has largest variability according to a well behaving measure. The divide and conquer algorithm is terminated when the estimated variability in all sub-domains are less than a preset limit. This approach could ideally produce an answer with guaranteed accuracy, but the measure we use for the variance contribution is only indicative. The variance contribution from a node is calculated by the following approach: Consider a sub-domain ω_i . Place one point p_0^i in the middle of ω_i and two points p_{j1}^i, p_{j2}^i on each line l_j going through the middle of ω_i , parallel with the axis A_j . The points $p_{j.}^i$ are located on each of the borders of ω_i . For each point, find the value $f(p_{j.}^i)$ of the integrand. This setup is illustrated in fig. 3 for a two dimensional example. We estimate the contribution to the variance from each sub-domain by the expression

$$\delta_i = \frac{|\omega_i|}{2D} \left(\sum_{j=1}^D \frac{f(p_{j1}^i) + f(p_{j2}^i)}{2} - f(p_0^i) \right), \quad (4)$$

where the volume of the sub-domain is calculated by $|\omega_i| = \prod_{j=1}^D (p_{j2}^i - p_{j1}^i)$. The axis we divide in order to create a finer partitioning is found by one of the expressions

$$A_i = \sup_j \{j; |f(p_{j2}^i) - f(p_0^i)| + |f(p_{j1}^i) - f(p_0^i)|\} \quad (5)$$

$$\text{or } A_i = \sup_j \{j; |f(p_{j2}^i) + f(p_{j1}^i) - 2f(p_0^i)|\} , \quad (6)$$

depending on the problem. These calculations are performed recursively as given in alg. 1. In [1] they use a fourth difference operator, but we find that the rule (5), which is a second difference operator, suits our setup better and gives better overall performance for the types of problems we have tested.

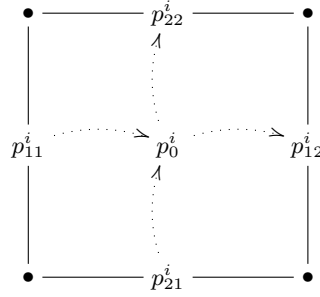


Figure 1. Illustration of the placement of points in ω_i in order to calculate the axis of division and the estimated variance of the sub-domain. Note that $\Omega = [0, 1]^2$ in this illustration

If the adaptive algorithm performs perfectly in accordance with the assumptions, the contribution from each sub-domain to the overall variance should be equal. Therefore $\sigma_i |\omega_i| = c, \forall i$ ideally. But even if the adaptive process aspires to use the simulation points as effectively as possible, we get some sub-domains in which the measured variability is close to the preset limit, and some where the variability is considerably lower than the limit. To circumvent this behavior we use less simulation points in the sub-domains where the variability is low. Theoretically, the fraction for the optimal allocation of points in each sub-domain can be shown to be

$$q_i^* = \frac{r_i \sigma_i}{\sum_{l=1}^P r_l \sigma_l}, \quad i = 1, \dots, P, \quad (7)$$

where r_i is the probability for a point to be contained in each bin represented by ω_i . Therefore $r_i = |\omega_i|$ in our setting. This leads to the allocation of simulation points by the relation

$$\Delta L_{i+1} = L \frac{\sigma_i |\omega_i|}{\sum_{l=1}^P \sigma_l |\omega_l|}, \quad (8)$$

where L is the total budget of simulation points. This approach, however, assume knowledge of all σ_i , and the adaptive approach does not provide this knowledge at the stage in the process where the contributions to the value of the total integral are calculated. Instead, we have chosen to implement a simpler approach to finding the number of simulation points in each bin. We use the relation

$$\Delta L_{i+1} = L \frac{\sigma_i}{\delta}, \quad (9)$$

where δ is the stopping criterion for the adaptive process on the variance estimates. This approach avoid the overhead by traversing the tree to collect the σ_i values, and in our tests the approach works well compared to using the same number of simulations in each bin.

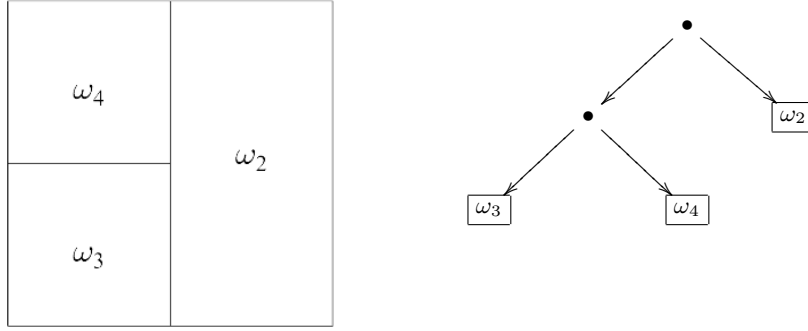


Figure2. Illustration of method for partitioning the domain $\Omega = \cup_i \omega_i$. The framed boxes in the tree cover the domain without overlapping

The adaptive approach is of course infested with overhead compared to plain simulation. For the adaptive method to be competitive with regards to computing time, one must focus on building good data structures and use effective algorithms for the administration of the simulation. In addition to the overhead connected to the creation and traversing of the binary tree holding the information about the segmentation of the domain, extra integrand evaluations must be performed in order to guide the partitioning of the domain. See fig. 2 for an illustration. We are unfortunately not able to use the calculated points in the adaptive process in the estimation of the integral, because a bias will be introduced. In many classes of problems, however, the domain can be reused when finding other parameters of the problem. An example of this can be found in finance where one often need to calculate the hedges (derivatives with respect to certain parameters) of an option contract as well as the price, see [2] on simulation of hedging parameters, and [11] and [12] for a formulation of the hedges by Malliavin calculus. In the adaptive setting the hedges can be calculated effectively without having to recreate the partitioning of the domain, see e.g. [10] or [7] for formulation and calculation of

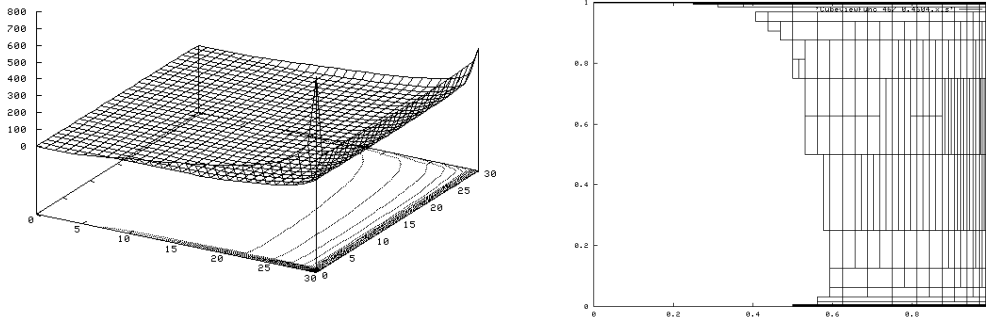


Figure3. An example (in the right part of the figure) of how the adaptive method would partition the domain for the payoff function of a two asset basket option graphed in the left figure

hedging parameters using this framework. The extra computer work involved in the adaptive method must therefore be accounted for in the comparison with the conventional method. In many classes of problems the adaptive method has a far better performance, even when all aspects of increased time consumption are taken into account. For a full discussion of the criteria on the problem for this to be the case, look at the companion paper [6].

In addition to simply sub-dividing the problem, there are several techniques that can be applied to the integrand in each sub-domain to improve the performance further. We notice, however, that the performance is not always increased. An investigation and discussion of these aspects are also carried out in the companion paper [6]. In the next section we present the techniques for variance reduction on the estimator of the integral, and focus on methods which can be applied to each sub-problem from the domain-decomposition in the adaptive process.

4 Local Variance Reduction

The estimator F for the unknown integral is sensitive to the choice of sampling points. This is the background for the possibility to find estimators and techniques delivering a low variance estimate. Variance reduction techniques may involve using calculation methods, such as QMC, designed to remove uncertainty. A supplementary approach often used together with the QMC principle, is to find alternative estimators for F than the mean value. The only requirement on the new estimator is that it has no, or neglectable, bias. Control variates (CV), weighted uniform sampling (WUS), stratified sampling (SS) and importance sampling (IS) are categories of techniques to provide such estimators. In our adaptive setting, we aim at employing such techniques also locally in each sub-domain. If we denote by G_ω the exact integral of a chosen function $g_\omega(x)$ in a sub-domain

ω , the estimator F_ω of the unknown integral in a local domain can be found by the modified estimators;

$$\text{CV: } F_\omega = \sum_{l=L_i}^{L_{i+1}} (f_\omega(y_l) - g_\omega(y_l)) + G_\omega \quad (10)$$

$$\text{WUS: } F_\omega = \frac{\sum_{l=L_i}^{L_{i+1}} f_\omega(y_l)}{\sum g_\omega(y_l)} G_\omega \quad (11)$$

$$\text{IS: } F_\omega = \sum_{l=L_i}^{L_{i+1}} \frac{f_\omega(z_l)}{g_\omega(z_l)} G_\omega, \quad (12)$$

where $y \sim \omega$ uniformly, while $z \sim \frac{g(y)}{G_\omega}$ (which is also in ω , but not uniformly). A description of how to generate z for a general g is given in e.g. [21]. Stratification is somewhat different than the techniques formulated in (10)-(12), and aims at distributing the arguments to f evenly, so as to guarantee that the average is actually representable for f . We explain the connection between SS and the adaptive method in the next section. In the following sections we handle CV and IS. WUS is very similar to CV and the same considerations apply. We therefore refrain from covering WUS in addition to CV.

4.1 Stratification

Stratification is a powerful technique and is related with the principles of the QMC method. The goal is to create a distribution of points in the domain for which a guaranteed fraction lies in specific bins. This is actually what is going on in special types of low discrepancy sequence generators such as the Latin hypercube sampling method. The adaptive method is an advanced sort of stratification in that we adapt the need for stratification as the integrand is changing. To gain further benefit from the stratification, it is for some problems possible to modify the integrand so that one can stratify the dimensions of the problem contributing the most to the variance of the integral estimator. In the option pricing problem one can re-engineer the problem so that the dimensions in the resulting integration problem correspond to independent stochastic processes in the original problem of finding an estimated expected value. By doing this, we can find the dimensions of the problem having the biggest contribution to the uncertainty of the final answer. By using more sampling points along the most important dimensions and by guaranteeing that these sampling points are evenly distributed, we can reduce the uncertainty, while keeping the number of sampling points low. Stratification is in the option pricing problem taken care of by the use of an SV-decomposition of the covariance matrix observed from the market or derived from the problem, and a rearranging of the independent noise components according to the size of the corresponding eigenvalues. When we then use a QMC method as the engine for

simulating the values, stratification is, depending of the low discrepancy sequence used, more or less taken care of implicitly. In addition to change the integration problem to be able to identify the biggest noise components, we have locally experimented with rearranging the noise components according to the axis along which the integrand changes most rapid, but this does not seem to give consistent gain in overall variance levels of the final estimator.

4.2 Control Variates

One method of achieving further variance reduction is to use the so called Control variate technique. We will employ this technique locally in each sub-domain. It must be adjusted to the limitations in number of integrand evaluations we need to impose on our selves to maintain speed, and we therefore use the integrand values in the points p_{j1}, p_{j2}, p_0 , which we have already calculated, to find a function $g_i(x)$ replicating the integrand f as closely as possible in each domain ω_i . This approach enables us to use the sampling points in the QMC approach to evaluate a modified integral with lower value, and thus smaller variance. The unbiased estimator of the integral in the domain ω_i is obtained by adding the known deterministic value G_i of the integral of the function g_i to the integral of the difference between f and g_i . Lets introduce some notation to state this exact: Pricing an option corresponds to evaluating the integral $\int_{[0,1]^D} f(x) dx$. In each sub-domain we evaluate $\int_{\omega_i} f(x) dx$. The idea is to find an approximate function $g_i(x)$ for $f(x)$ in each sub-domain ω_i , having the property that the integral G_i of g_i in this domain can be found deterministic. That is,

$$F_i = \int_{\omega_i} f(x) dx \tag{13}$$

$$= \int_{\omega_i} f(x) - g_i(x) dx + G_i . \tag{14}$$

If the integral of the difference between the functions f and g_i is smaller than the integral of f in ω_i , this approach will give an estimate of F_i with lower variance than evaluating the integral of $f(x)$ in ω_i directly.

Finding g_i and G_i have to be relatively cheap for this approach to be competitive. We have developed three different function classes that meet these demands. In the following we drop the subscripts i and ω_i on the functions for cleaner notation. Note, however, that all parameters, functions and integrals in the remaining part of the section are found for a general local domain. The three function classes

are given by

$$g(x) = \sum_{j=1}^D a_j (x_j - h_j)^{e_1} + \sum_{j=1}^D b_j (x_j - h_j)^{e_2} + c \quad (15)$$

$$g(x) = \exp\left(\sum_{j=1}^D a_j (x_j - h_j)\right) - \exp\left(\sum_{j=1}^D b_j (x_j - h_j)\right) + c \quad (16)$$

$$g(x) = \sum_{j=1}^D a_j (\exp(b_j (x_j - h_j)) - 1) + c, \quad (17)$$

where $h_j = (p_{j1} + p_{j2})/2$. These functions are accommodating with regard to the placement of the points in the sub-domains of the adaptive algorithm, since we as part of the adaptive process already have evaluated f in these points. To find the parameters a_j, b_j and c we only need to solve D equations in two variables and one equation for c . First we consider the polynomial function in (15). The solution of the equations for $a_j, b_j \forall j$ for this function has the structure:

$$a_j = \frac{\bar{h}_j^{e_2} (f(d_j) - c) - (-\bar{h}_j)^{e_2} (f(u_j) - c)}{\bar{h}_j^{e_1} (-\bar{h}_j)^{e_2} - \bar{h}_j^{e_2} (-\bar{h}_j)^{e_1}} \quad (18)$$

$$b_j = \frac{(-\bar{h}_j)^{e_1} (f(d_j) - c) + \bar{h}_j^{e_1} (f(u_j) - c)}{\bar{h}_j^{e_1} (-\bar{h}_j)^{e_2} - \bar{h}_j^{e_2} (-\bar{h}_j)^{e_1}}, \quad (19)$$

where $\bar{h}_j = (p_{j2} - p_{j1})/2$ and $c = f(h_1, h_2, \dots, h_D)$. The value of the integral of g over a sub-domain is

$$G = \prod_{k=1}^D 2(\bar{h}_k) \left(\frac{1}{e_1 + 1} \sum_{j=1}^D \frac{a_j ((\bar{h}_j)^{e_1+1} - (-\bar{h}_j)^{e_1+1})}{2\bar{h}_j} + \frac{1}{e_2 + 1} \sum_{j=1}^D \frac{b_j ((\bar{h}_j)^{e_2+1} - (-\bar{h}_j)^{e_2+1})}{2\bar{h}_j} + c \right). \quad (20)$$

It is not difficult to see that if $(e_1 + 1)$ is divisible by two, the first part does not contribute to the integral, and similarly for the second part if $(e_2 + 1)$ is divisible by two. The relation between e_1 and e_2 must be controlled in order for the equations to have solution: Without losing generality assume $a_1 \geq a_2$. We must have $a_1 = a_2 + (2n + 1)$, where $n \in \{0, 1, \dots\}$.

The class of functions where we have a sum of exponential functions, given in (16), is also rather easy to handle: Let $r_{j1} = f(p_{j1}) - c$, $r_{j2} = f(p_{j2}) - c$ and $\bar{h}_j = (p_{j2} - p_{j1})/2$. Then the parameters of this class are given by

$$a_j = -\frac{r_{j1} r_{j2}}{r_{j2} + r_{j1}} \quad b_j = \frac{\ln(-r_{j1}/r_{j2})}{\bar{h}_j}, \quad (21)$$

and the value of the integral is

$$G = \left(\prod_{j=1}^D 2\bar{h}_j \right) \left[\sum_{j=1}^D a_j \left(\frac{\exp(b_j \bar{h}_j) - \exp(-b_j \bar{h}_j)}{2b_j \bar{h}_j} - 1 \right) + c \right]. \quad (22)$$

It is, however, evident that this function class is not very adapted to the option pricing problem in general, because the sum of variables in the argument to the exponential function is not taken into account. If we look at the last function class, given in (17), this is taken better care of. We can find the parameters of this class by solving a set of equations as before. Using the same notation as above, the parameters are

$$a_j = \frac{\ln\left(\frac{1}{2r_{j1}}(r_{j1}r_{j2} \pm \sqrt{r_{j1}^2 r_{j2}^2 - 4r_{j1}r_{j2}})\right)}{\bar{h}_j} \quad (23)$$

$$b_j = \frac{\ln\left(-r_{j2} + \frac{1}{2r_{j1}}(r_{j1}r_{j2} \pm \sqrt{r_{j1}^2 r_{j2}^2 - 4r_{j1}r_{j2}})\right)}{\bar{h}_j}, \quad (24)$$

where we pick the root which ensures that we get a positive argument to the $\ln(\cdot)$ function. Note however, that we are not able to find a solution to all a_j, b_j for general f , and the function class therefore is not sufficiently robust to be used alone. To be employed in applications, it must be combined with one of the other classes for the dimensions resulting in a negative root as argument to the $\ln(\cdot)$ function. The value of the integral of the function in (17) is given by

$$G = \prod_{j=1}^D \frac{\exp(a_j \bar{h}_j) - \exp(-a_j \bar{h}_j)}{a_j} - \prod_{j=1}^D \frac{\exp(b_j \bar{h}_j) - \exp(-b_j \bar{h}_j)}{b_j} + c \prod_{j=1}^D 2\bar{h}_j. \quad (25)$$

The function class capturing the mix of the two exponential function classes, (16) and (17), is given by

$$g(x) = \exp\left(\sum_{j \in \hat{D}} a_j(x_j - h_j)\right) - \exp\left(\sum_{j \in \hat{D}} b_j(x_j - h_j)\right) + \sum_{j \notin \hat{D}} a_j(\exp(b_j(x_j - h_j)) - 1) + c, \quad (26)$$

where $\hat{D} \subset \{1, \dots, D\}$ is the set of dimension where the root is positive. The parameters are given by (23) and (24) for respectively a_j and b_j for $j \in \hat{D}$, and by (21) for $j \in \{1, \dots, D\} \setminus \hat{D}$. c is equal to the value of f in the mid-point of

the domain. The integral is the sum of the expressions given in (22) and (25) for the appropriate sets of indices.

As a closure of this section I will briefly mention aspects concerning a common control variate technique in Asian option pricing; namely to use the geometric average Asian option as a control variate (for which there exists an easier calculation routine) when pricing the arithmetic average Asian option. In the adapted routine this would not serve as a local variance reduction technique, because it would not take advantage of the gathered information about the local behavior. Instead, the parameters used throughout would be static, amounting to alter the problem only in a global fashion.

4.3 Importance sampling

We have been looking at the possibility to perform importance sampling locally for each sub-domain by finding general distribution functions suited for this approach. As we shall see however, the attempts turned out to produce functions which were too hard to calculate as part of the adaptive process. Therefore we have only implemented a version in which the importance sampling is applied to the globally defined function before the adaptive procedure is started. The main result of these calculations with regards to the adaptive method, is that the adaptive method performs equally well together with importance sampling compared to the non-adaptive method. It is, however, not always improvements tied to the use of either importance sampling nor the adaptive method. For details on these results, consult the companion paper [6]. References on importance sampling for finance problems are [15], [14], [18]. The first part of this section investigates local IS, and show that it is difficult to apply.

In order to formulate the inverse cumulative distribution function appearing in the general IS framework, we use the approach described in [21] on the basis of the function classes (16) and (17). Let y be the low discrepancy sequence in $[0, 1]^D$, and let $g(x_1, \dots, x_D)$ be a probability density function and $G(x_1, \dots, x_D)$ its distribution function. Denote by $g_{1,\dots,k}(x_1, \dots, x_k)$ the marginal density function for x_1, \dots, x_k , in general given by

$$g_{1,\dots,k}(x_1, \dots, x_k) = \int_{[0,1]^{D-k}} g(x_1, \dots, x_D) dx_{k+1} \dots dx_D . \quad (27)$$

The distribution function $G_k(x_k)$ associated with the conditional density for x_k given x_1, \dots, x_{k-1} , is then given by

$$G_k(x_k) = \int_0^{x_k} \frac{g_{1,\dots,k}(x_1, \dots, x_k)}{g_{1,\dots,k-1}(x_1, \dots, x_{k-1})} dx . \quad (28)$$

To obtain a g -distributed sequence z in $[0, 1]^D$, we need to find the function $G_k^{-1}(\cdot)$. Then $z_n = (z_{n1}, \dots, z_{nD})$ where $z_{ni} = G_i^{-1}(y_{ni})$. This principle can theoretically

be applied to each sub-domain by changing the function $g(x_1, \dots, x_D)$ to mimic the function f in each domain. For the function class (17), the integration in (27), without the integration limits, yields

$$g_{1,\dots,k}(x_1, \dots, x_k) = \sum_{i=1}^k a_i \exp(b_i(x_i - h_i)) \prod_{j=k+1}^D x_j \\ + \sum_{i=k+1}^D \left[\frac{a_i}{b_i} \exp(b_i(x_i - h_i)) \prod_{\substack{j=k+1 \\ j \neq i}}^D x_j \right] + \left(c - \sum_{i=1}^D a_i \right) \prod_{j=k+1}^D x_j, \quad (29)$$

where D is the total dimension of the integrand. When the integration limits are inserted for the local domain, and the expression for $G_k(x_k)$ is calculated, we get a complicated expression in terms of the constants. This expression can, however, be reduced to a rather tangible function in one variable when the constants are collected, giving

$$G_k(x_k) = c_{k1} \exp(c_{k2}x_k - c_{k3}) + c_{k4}x_k, \quad (30)$$

where c_{ki} are constants. We therefore have to solve an equation of the form (30) to find the g -distributed sequence $\{z_l\}$ for this class of functions. The solution is

$$x_k = a_{k1}W(a_{k2} \exp(a_{k3}y + a_{k4})) + a_{k5}, \quad (31)$$

where W is the so called Lambert W function and a_{ki} are constants. References on the Lambert W function are e.g. [5] and [4].

In a similar manner we can develop the distribution function for the class given in eq. (16);

$$g_{1,\dots,k}(x_1, \dots, x_k) = \frac{\exp(\sum_{i=1}^s a_i(x_i - h_i))}{\prod_{i=k+1}^s a_i} \\ - \frac{\exp(\sum_{i=1}^s b_i(x_i - h_i))}{\prod_{i=k+1}^s b_i} + c \prod_{i=k+1}^s x_i, \quad (32)$$

giving

$$G_k(x_k) = c_{k1} \exp(c_{k2}x_k - c_{k3}) - c_{k4} \exp(c_{k5}x_k - c_{k6}) + c_{k7}, \quad (33)$$

where c_{ki} are general constants. The inversion of this function is given by

$$x_k = a_{k1}R(a_{k2} \exp(Z) + a_{k3} \exp(a_{k4}Z + a_{k5}) + a_{k6} + a_{k7}y) + a_{k8}, \quad (34)$$

where the function R means that we have to find the roots with regards to Z of the argument to R .

Both the Lambert W function and the root in (34) can probably be implemented, but would be rather slow to use in the adaptive setting as they require numerical estimation or series expansion approximation. I have also developed the $G^{-1}(\cdot)$ function for the class of polynomial functions described in eq. (15). In order to apply the IS technique with this function class as a base, we need to find the roots of a polynomial of degree $(e1 + 1)$, and in order for the method to be effective we need $e1 > 1$. The consequence is that we have to find the roots of at least a cubic polynomial.

The previous discussion shows that it is hard to find a general function class suited for the IS technique together with the adaptive method. An alternative is to try to change the measure under the constraint that distribution is still normal, but with other parameters. This amounts to choosing the function g to be a version of the normal distribution function. Following the deduction in e.g. [15] we look at functions $g(z) > 0 \Rightarrow h(z) > 0, \forall z \in \Omega$. Then

$$E_g[G(Z)] = \int_{\Omega} G(z)g(z)dz \quad (35)$$

$$= \int_{\Omega} G(z) \frac{g(z)}{h(z)} h(z)dz \equiv E_h[G(Z) \frac{g(Z)}{h(Z)}], \quad (36)$$

where the subscript of the expectation indicates which measure it is taken under. The factor $g(Z)/h(Z)$ is the Radon-Nikodym derivative. The function h now gives a degree of freedom with respect to minimizing the variance.

$$\int_{\Omega} \left(G(z) \frac{g(z)}{h(z)} - E_g[G(Z)] \right)^2 h(z)dz. \quad (37)$$

In our setting g is a normal distribution, and we want h to be a normal distribution as well, with the same variance structure as g . The importance sampling estimator then becomes

$$E_g[G(Z)] = E_h[G(Z) \exp(-\mu^T Z + \frac{1}{2}\mu^T \mu)] \quad (38)$$

$$= E_g[G(Z + \mu) \exp(-\mu^T Z - \frac{1}{2}\mu^T \mu)]. \quad (39)$$

In [15] a detailed discussion of constructing the optimization problem for finding the best μ for h is presented. In [16] a simpler interpretation is given. It states that if we let $G(z) \equiv \exp(F(z))$, and F is approximately linear near μ , then $F(Z + \mu)$ in (39) is approximately $F(\mu) + \nabla F(\mu)Z$. The substitution yields

$$\begin{aligned} \exp(F(Z + \mu) - \mu^T Z - \frac{1}{2}\mu^T \mu) &\approx \\ \exp(F(\mu) + \nabla F(\mu)Z - \mu^T Z - \frac{1}{2}\mu^T \mu). \end{aligned} \quad (40)$$

In order to make this expected value a non-stochastic variable we need to ensure that

$$\nabla F(\mu)Z - \mu^T Z \equiv 0 \quad (41)$$

$$\Rightarrow \nabla F(\mu) = \mu . \quad (42)$$

The solution to (42) can under conditions stated in [15] be found from a fix-point equation. We limit ourselves to stating that the method is working in the setting of our test problems. Numerical results are presented in the accompanying article [6].

Acknowledgements

The author is grateful for financial support from Storebrand Investments. I also want to thank F. E. Benth for valuable talks and for applying amendments to the manuscript.

Appendix A

Algorithm 1: Adaptive subdivision

Description: The basic element of the algorithm is a node $N\omega$ representing a sub-domain. $N\omega$ has one left and one right pointer which are able to point to other nodes. Each node know its volume, can hold an estimate of the contributed variance, and knows the coordinates of its corners.

Input: A pointer $\text{ptr}N\omega$ pointing to a node $N\omega$

Output: A binary tree where the leaf nodes cover the domain exactly. The sum of the values in the leaf nodes is the value of the integral.

DC($\text{ptr}N\omega$)

if FINDAXISANDVARIANCE($\text{ptr}N\omega$) > MaxVar

 CREATE($\text{ptr}N\omega.\text{Left}$)

 DC($\text{ptr}N\omega.\text{Left}$)

 CREATE($\text{ptr}N\omega.\text{Right}$)

 DC($\text{ptr}N\omega.\text{Right}$)

References

1. J. Berntsen, T. O. Espelid, and A. Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software*, 1991.
2. M. Broadie and P. Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42:269–285, 1996.
3. R. Cools. The approximation of low-dimensional integrals: available tools and trends. Technical report, Department of Computer Science, K.U.Leuven, May 1997.
4. R. M. Corless, , D. J. Jeffrey, and D. E. Knuth. A sequence of series for the Lambert w function. In W. Küchlin, editor, *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, pages 197–204, 1997.

5. R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On Lambert w function. *Advances in Computational Mathematics*, 5:329–359, 1996.
6. L. O. Dahl. An adaptive method for evaluating multidimensional contingent claims. Part II. Accepted in the 8th International Vilnius Conference on Probability Theory and Mathematical Statistics, June 2002. In preprint at http://www.math.uio.no/eprint/pure_math/2002/pure_2002.html, February 2002.
7. L. O. Dahl. Fast evaluation of derivatives of the Asian basket option by SVD and adaptive integration. Manuscript, April 2002.
8. L. O. Dahl and F. E. Benth. Valuation of Asian basket options with quasi-Monte Carlo techniques and singular value decomposition. Submitted to Journal of Computational Finance. Currently in preprint at http://www.math.uio.no/eprint/pure_math/2001/pure_2001.html, February 2001.
9. L. O. Dahl and F. E. Benth. Fast evaluation of the Asian basket option by singular value decomposition. In K.-T. Fang, F.J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, 2002.
10. L. O. Dahl, F. E. Benth, and K. H. Hvistendal. On derivatives of claims in commodity and energy markets using a Malliavin approach. In preprint at http://www.math.uio.no/eprint/pure_math/2002/pure_2002.html, April 2002.
11. E. Fournié, J. M. Lasry, J. Lebuchoux, P. L. Lions, and N. Touzi. Application of malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics*, 3:391–412, 1999.
12. E. Fournié, J. M. Lasry, J. Lebuchoux, P. L. Lions, and N. Touzi. Application of malliavin calculus to Monte Carlo methods in finance ii. *Finance and Stochastics*, 5:201–236, 2001.
13. A. Genz and R. Cools. An adaptive numerical cubature algorithm for simplices. Technical report, Department of Computer Science, K.U.Leuven, December 1997.
14. P. Glasserman, P. Heidelberger, and P. Shahabuddin. Gaussian importance sampling and stratification: Computational issues. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, editors, *Proceedings of the 1998 Winter Simulation Conference*, volume 1, pages 685–693. IEEE Computer Society Press, September 1998.
15. P. Glasserman, P. Heidelberger, and P. Shahabuddin. Asymptotically optimal importance sampling and stratification for pricing path-dependent options. *Mathematical Finance*, 9(2):117–152, 1999.
16. P. Glasserman, P. Heidelberger, and P. Shahabuddin. Importance sampling in the Heath-Jarrow-Morton framework. *Journal of Derivatives*, 7(1):32–50, 1999.
17. P. Glasserman, P. Heidelberger, and P. Shahabuddin. Importance sampling and stratification for value-at-risk. In *Computational Finance 1999 (Proceedings of the Sixth International Conference on Computational Finance)*, pages 7–24. MIT Press, 2000.
18. Kloeden and Platon. *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Application of Mathematics, Stochastic Modelling and Applied Probability*. Springer Verlag, 1995.
19. L. Kocis and W. J. Whiten. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2):266–294, June 1997.
20. H. Niederreiter and C. Xing. The algebraic-geometry approach to low-discrepancy sequences. In *Monte Carlo and Quasi-Monte Carlo Methods 1996*, number 127 in Lecture Notes in Statistics, pages 139–160. Springer Verlag, 1996.
21. G. Ökten. Error reduction techniques in quasi-Monte Carlo integration. Downloadable, 1996 (?).
22. A. Owen and Y. Zhou. Safe and effective importance sampling. *Journals of the American Statistical Association*, 95(449):135–143, 2000.
23. A. B. Owen. Latin supercube sampling for very high dimensional simulations. *ACM Transactions on Modeling and Computer Simulation*, 8(1), 1998.
24. A. B. Owen. Monte Carlo extension of quasi-Monte Carlo. Technical report, Department of Statistics, Stanford University, 1999.
25. W. H. Press and G. R. Farrar. Recursive stratified sampling for multidimensional Monte Carlo integration. *Computers in Physics*, 1990.
26. F. J. Vázquez-Abad and D. Dufresne. Accelerated simulation for pricing Asian options. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, editors, *Proceedings of the 1998 Winter Simulation Conference*, 1998.